

Lecture Notes on Adjoint Sax

15-417/817: HOT Compilation
Frank Pfenning

Lecture 14
March 11, 2025

1 Introduction

The target of compilation from adjoint ND will be adjoint Sax. It turns out that adjoint Sax is actually quite a bit simpler than adjoint ND for two reasons: (1) because of the symmetries between positive and negative types afforded by Sax we need fewer new constructs, and (2) any kind of overloading of definitions with multiple types or modes are resolved during compilation so that Sax (as a lower-level language) can be monomorphic. Also, we can reuse the definitions of operations such as merge and least upper bound from [Lecture 10](#) without change.

2 The Usual Connectives

We show the usual rules without taking into account subtyping.

$$\frac{y : A_m \in \Gamma}{\Gamma \vdash \text{id } x \ y :: (x : A_m) / (y : A_m)} \text{id}$$
$$\frac{\Gamma \vdash P :: (x : A_m) / \Xi_1 \quad (m \geq r) \quad \Gamma, x : A_m \vdash Q :: (d : D_r) / \Xi_2}{\Gamma \vdash \text{cut } (x : A_m) \ P \ Q :: (d : D_r) / \Xi_1 ; (\Xi_2 \setminus x_m)} \text{cut}$$

In the cut rule we are reminded that in Sax, too, the independence principle should hold. That is, in $\Gamma \vdash P :: (d : D_r) / \Xi$ then Γ is the lexical context and Ξ are the variables used in P where $\Xi \geq r$ must be respected. In the cut rule, we know from the first premise that $\Xi_1 \geq m$, and also from the middle condition that $m \geq r$, and from the last premise that $\Xi_2 \geq r$. So $\Xi_1 ; \Xi_2 \geq r$ as required.

Pairs $A_m \otimes B_m$.

$$\frac{a : A_m \in \Gamma, b : B_m \in \Gamma}{\Gamma \vdash \text{write } c \ (a, b) :: (c : A_m \otimes B_m) / (a : A_m) ; (b : B_m)} \otimes X$$
$$\frac{c : A_m \otimes B_m \in \Gamma \quad (m \geq r) \quad \Gamma, x : A_m, y : B_m \vdash P :: (d : D_r) / \Xi}{\Gamma \vdash \text{read } c \ ((x, y) \Rightarrow P) :: (d : D_r) / (\Xi \setminus x_m \setminus y_m) ; (c : A_m \otimes B_m)} \otimes L$$

Here, in the axioms it is important to join $(a : A_m)$ and $(b : B_m)$ because if mode m admits contraction then a and b could be the same. Also, in the $\otimes L$ rule we need to check that $m \geq r$

to ensure the resulting output context is also above r . Recall that the context removal operation requires the mode of the variables because if m allows weakening it need not be present in Ξ .

Functions $A_m \rightarrow B_m$.

$$\frac{\Gamma, x : A_m \vdash P :: (y : B_m) / \Xi}{\Gamma \vdash \mathbf{write} \ c \ ((x, y) \Rightarrow P) :: (c : A_m \rightarrow B_m) / (\Xi \setminus x_m)} \rightarrow R$$

$$\frac{a : A_m \in \Gamma, c : A_m \rightarrow B_m \in \Gamma}{\Gamma \vdash \mathbf{read} \ c \ (a, b) :: (b : B_m) / (a : A_m) ; (c : A_m \rightarrow B_m)} \rightarrow L$$

Independence doesn't require any additional checks. But we need to be careful to use join in the output of $\rightarrow L$ because with our equirecursive types, a and c could actually be the same address.

Unit 1.

$$\frac{}{\Gamma \vdash \mathbf{write} \ c \ () :: (c : \mathbf{1}_m) / ()} \mathbf{1}X \quad \frac{c : \mathbf{1}_m \in \Gamma \quad (m \geq r) \quad \Gamma \vdash P :: (d_r : D_r) / \Xi}{\Gamma \vdash \mathbf{read} \ c \ (() \Rightarrow P) :: (d_r : D_r) / \Xi ; (c : \mathbf{1}_m)} \mathbf{1}L$$

Sums $\oplus \{\ell : A_m^\ell\}_{\ell \in L}$. For the sum type we need the least-upper-bound operation on output contexts. This is well-defined because there must be at least one branch because sums must be nonempty.

$$\frac{a : A_m^k \in \Gamma}{\Gamma \vdash \mathbf{write} \ c \ k(a) :: \oplus \{\ell : A_m^\ell\}_{\ell \in L} / (a : A_m^k)} \oplus X$$

$$\frac{c : \oplus \{\ell : A_m^\ell\}_{\ell \in L} \in \Gamma \quad m \geq r \quad (\Gamma, x^\ell : A_m^\ell \vdash P^\ell :: (d : D_r) / \Xi^\ell) \quad (\forall \ell \in L)}{\Gamma \vdash \mathbf{read} \ c \ \{\ell(x^\ell) \Rightarrow P^\ell\}_{\ell \in L} :: (d : D_r) / \bigsqcup_{\ell \in L} (\Xi^\ell \setminus x_m^\ell) ; (c : \oplus \{\ell : A_m^\ell\}_{\ell \in L})} \oplus L$$

Lazy Records $\& \{\ell : A_m^\ell\}_{\ell \in L}$. As for the other dual connectives, we use the syntax for sums, but we swap the read and write operations.

$$\frac{(\Gamma \vdash P^\ell :: (x^\ell : A_m^\ell) / \Xi^\ell) \quad (\forall \ell \in L)}{\Gamma \vdash \mathbf{write} \ c \ \{\ell(x^\ell) \Rightarrow P^\ell\}_{\ell \in L} :: (c : \& \{\ell : A_m^\ell\}_{\ell \in L}) / \bigsqcup_{\ell \in L} \Xi^\ell} \& R$$

$$\frac{c : \& \{\ell : A_m^\ell\}_{\ell \in L} \in \Gamma}{\Gamma \vdash \mathbf{read} \ c \ k(a) :: (a : A_m^k) / (c : \& \{\ell : A_m^\ell\}_{\ell \in L})} \& X$$

3 The Shift Connectives

We fully expect there to be a single new syntax that unifies the upshift and downshift at the level of programs since they are dual to each other. This is indeed the case. As for the others, the notation arises from the positive connective and the dual negative one borrows it.

Downshift $\downarrow_m^k A_k$. Because the downshift is positive, the right rule becomes an axiom, while the left rule remains essentially intact.

$$\frac{a : A_k \in \Gamma}{\Gamma \vdash \mathbf{write} \ c \ \langle a \rangle :: (c : \downarrow_m^k A_k) / (a : A_k)} \downarrow X$$

$$\frac{c : \downarrow_m^k A_k \in \Gamma \quad (m \geq r) \quad \Gamma, x : A_k \vdash P :: (d : D_r) / \Xi}{\Gamma \vdash \mathbf{read} \ c \ (\langle x \rangle \Rightarrow P) :: (d : D_r) / (\Xi \setminus x_k); (c : \downarrow_m^k A_k)} \downarrow L$$

In $\downarrow L$, we know by invariant on the premise that $\Xi \geq r$. The condition $m \geq r$ is needed so that $((\Xi \setminus x_k); (c : \downarrow_m^k A_k)) \geq r$. By presupposition we also know that $k \geq m$ and therefore $k \geq r$, although it is not entirely clear why we might need that for this rule.

Upshift $\uparrow_n^m A_n$.

$$\frac{\Gamma \vdash P :: (x : A_n) / \Xi \quad (\Xi \geq m)}{\Gamma \vdash \mathbf{write} \ c \ (\langle x \rangle \Rightarrow P) :: (c : \uparrow_n^m A_n) / \Xi} \uparrow R$$

$$\frac{c : \uparrow_n^m A_n \in \Gamma}{\Gamma \vdash \mathbf{read} \ c \ \langle a \rangle :: (a : A_n) / (c : \uparrow_n^m A_n)} \uparrow X$$

Keeping in mind the presupposition that $m \geq n$ when we write $\uparrow_n^m A_n$, we see that $\uparrow R$ perhaps surprisingly requires a condition while $\uparrow X$ requires none.

4 Call

As usual, we think of top-level functions as metavariables that abstract over a whole context. We assume all definitions $p \ (x : A_m) \ [\Delta] = P$ are collected in a global signature and we check each such definition independently. Here, $x : A_m$ is the destination of P .

$$\frac{(\Delta \geq A_m) \quad \Delta \vdash P :: (x : A_m) / \Xi \quad \Xi \setminus \Delta = \Xi'}{p \ (x : A_m) \ [\Delta] = P \text{ valid}}$$

Technically, the condition $\Delta \geq A_m$ could be weakened to $\Xi \geq m$, but it seems strange to parameterize a metavariable by variables that cannot be used in their definition.

When using metavariables, we supply a substitution ρ for its parameters.

$$\frac{p \ (x : A_m) \ [\Delta] = P \quad \Gamma \vdash \rho :: \Delta / \Xi}{\Gamma \vdash \mathbf{call} \ p \ [\rho] :: (a : A_m) / \Xi} \text{call}$$

$$\frac{}{\Gamma \vdash (\cdot) :: (\cdot) / (\cdot)} \quad \frac{\Gamma \vdash \rho :: \Delta / \Omega \quad b : B_m \in \Gamma}{\Gamma \vdash (\rho, y \mapsto b) :: (\Delta, y : B_m) / \Omega; (b : B_m)}$$

5 Subtyping

The rules for subtyping are extended in the expected covariant form to encompass the shifts, where the bounds have to match exactly. We only compare $A_m \leq B_m$, never two types of different modes.

Subtyping applies in a number of rules of Sax: essentially whenever a type appears more than once. For example, the rule

$$\frac{a : A_m \in \Gamma, b : B_m \in \Gamma}{\Gamma \vdash \mathbf{write} \ c \ (a, b) :: (c : A_m \otimes B_m) / (a : A_m) ; (b : B_m)} \otimes X$$

is modified to

$$\frac{a : A'_m \in \Gamma, b : B'_m \in \Gamma \quad A'_m \leq A_m \quad B'_m \leq B_m}{\Gamma \vdash \mathbf{write} \ c \ (a, b) :: (c : A_m \otimes B_m) / (a : A'_m) ; (b : B'_m)} \otimes X$$

Similarly, to account for width subtyping, some fields of lazy records or branches of continuations may be dropped. For example:

$$\frac{(K \supseteq L) \quad (\Gamma \vdash P^\ell :: (x^\ell : A_m^\ell) / \Xi^\ell) \quad (\forall \ell \in L)}{\Gamma \vdash \mathbf{write} \ c \ \{\ell(x^\ell) \Rightarrow P^\ell\}_{\ell \in K} :: (c : \&\{\ell : A_m^\ell\}_{\ell \in L}) / \bigsqcup_{\ell \in L} \Xi_\ell} \&R$$

As we will see in a future lecture, there are formulations of bidirectional typing for Sax which eliminates some of these subtype checks.